Final Project

EELE 467

Due date: 12/11/2024

The final project involves creating a system that can change the color of the color LED by turning potentiometers associated with each RGB color as illustrated in the system diagram below. You will also need to propose one additional hardware piece and one additional software piece per group member; that is, each group member needs a unique hardware piece and software piece, which interact with each other.

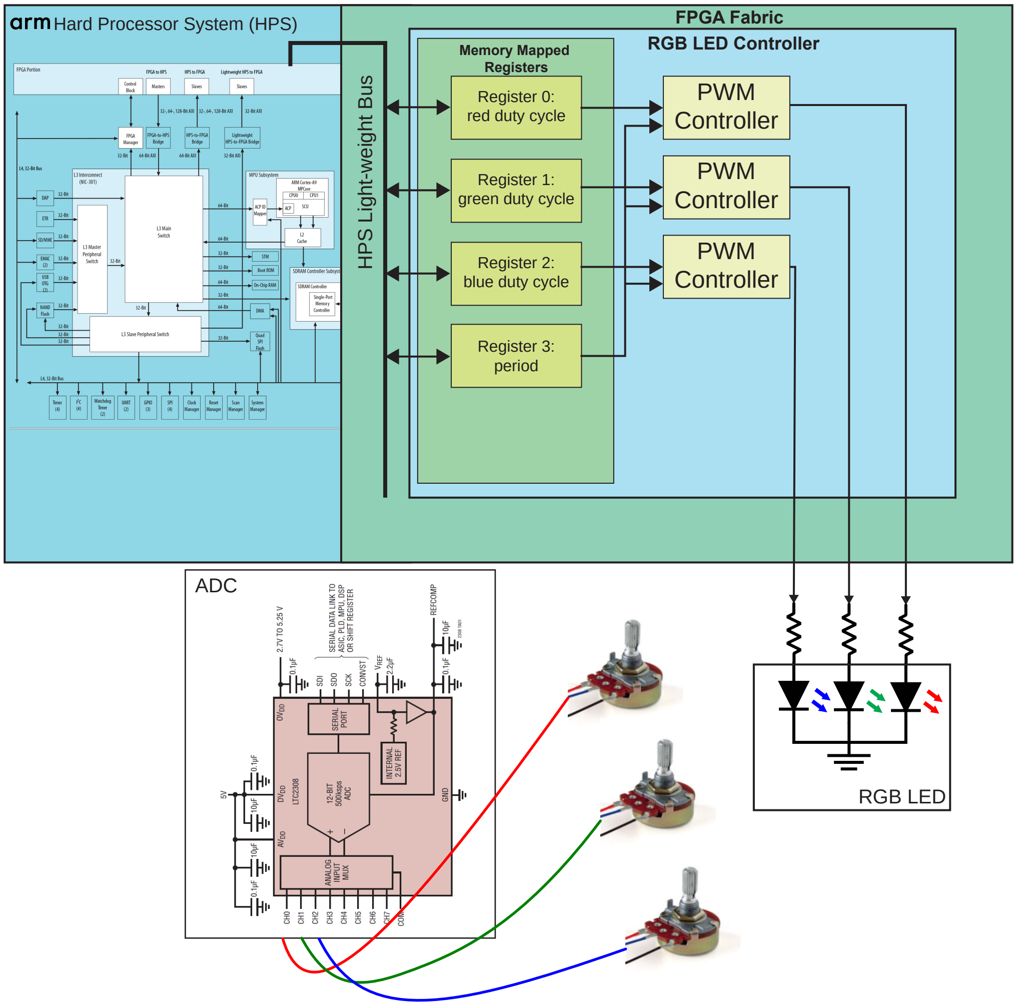

Figure 1: The color LED system will change the color of the LED based on the potentiometer settings read by the ADC on the DE10-Nano board.

Hardware

You will need the following hardware components. Make sure that your hardware works before moving on to the software section!

Hardware 1: The **RGB LED controller** that you created for HW 10.

Hardware 2: An **ADC Controller** to measure the voltage from three potentiometers as shown in the figure above. This ADC already exists on the DE10 Nano. You won't need to create VHDL code yourself. You will either have to instantiate an existing component in Platform Designer, or you will route some of the ADC control signals to the HPS; this is yet to be determined.

Hardware 3: **For each group member, create hardware that is unique to you.** Your IP core that interfaces to your hardware must have at least one register that is controlled from your custom software.
Possibilities include:

- Add a push button.
- Add a rotary encoder.
- Add a LED array
- Another Potentiometer.

Software

You will need to write two Platform Device Drivers, a C program, and a shell script.

Software 1: **Write a Platform Device Driver for RGB LED controller** that exports the registers to **sysfs**. Do this in a similar manner to Lab 10 (Device Tree) and Lab 11 (Platform Driver).

> ℹ The data types for your registers need to remain the same as what was individually assigned in Homework assignments. This means that the data types in the Platform Driver for the show() and store() functions need to be consistent with these data types.

Software 2: **Write a C program** that performs an infinite busy loop where it reads the potentiometer values, performs the appropriate conversions and scalings, and writes the control words to the associated color PWM components.

Software 3: **Write a device driver for your custom hardware**.

Software 4: **For each group member, create software that is unique to you and that uses your proposed hardware.** Possibilities include:

- Write a program that cycles through the spectrum of colors on the RGB LED, e.g., cycle through all hue values in an HSV color model: FastLED-HSV-Colors
- Use a button for something.
- Show the relative potentiometer value on the LED array.
- Write a program takes in hex color codes and outputs the corresponding color on the LED.

- Use the switches and LED array to show the values of the R, G, and B channels (scale the duty cycle to [0,255] and display the binary value on the LED array).

Teamwork Details

This project will be done in groups of two. Students will select their own partners.

The team members will need to collaborate on their hardware and software implementations. Each member will implement their own hardware; however, the team members will implement software for the *other* person's hardware.

Requirement 1: Each team member implements their own hardware.

Requirement 2: Each team member implements the device driver for the other person's hardware.

Requirement 3: Each team member performs a *code review* for the other person's VHDL and device driver.

Requirement 4: The division of labor must be included in the project proposal.

Hardware and Software Proposals

The team must propose two sets of hardware/software components. Each set can be completely independent of the other, or they can interact somehow. However, the hardware and software within a set must interact with each other.

The steps to get your hardware and software approved are:

Step 1: Check the Final Project folder on D2L to see if the idea you are thinking of has already been taken. Proposals will be approved on a first-come first-serve basis. If your idea has already been taken, you can propose a twist on the same idea. The proposals just can't be identical.

Step 2: Create a `proposal` branch in your final project repo.

Step 3: In your *final-project* repository in the \docs folder, create your proposal in the document **proposal.md**

Step 4: Create a pull request to get the proposal approved. In your repository, near the top, click on the *Pull requests* menu/tab. Then click on the green **New pull request** button. In your request, select tvannoy as the reviewer.

Step 5: Get approval for the proposal. If your proposal is not approved, you will need to resubmit until it is approved.

> ❗ Proposals are due by 11/15 at midnight!

Proposal Format

The `docs/proposal.md` file in your final project repository already contains a template for you to use. Please follow that template.

Extra Credit Available

There is 10% of extra credit available that is split 5% hardware and 5% software. If you are interested in getting extra credit for the final project, add a section to **proposal.md** that is clearly marked *Extra Credit* and propose what the extra credit tasks are.

Grading 💯

50% – Demo
25% – Code quality
25% – Documentation / Report

GitHub Repository

1. Accept the GitHub Classroom assignment: https://classroom.github.com/a/CBsdFujW
2. The first team member to accept the assignment will need to create a team name. Please name the team as firstname1-firstname2.

Final Project Deliverables

You will need the following items submitted/demoed by end of day Wednesday December 11.

Project Demonstration

You will need to demonstrate your final project in person. The location for demos will be in the Digital Lab (COBH 601). Your project needs to be demonstrated where you show your system working, i.e., you can turn the potentiometers and change the color on the color LED. Show the changing the LED color via software as well as doing it on the command line. You will need to demonstrate your custom proposal as well. You will need to explain how your code works.

You will need to make an appointment via MSU's Appointment Scheduler for a 30-minute time slot for the project demo and to discuss your project/code. To make an "advising" appointment, please go to final project appointment scheduler. You don't have to wait until finals week to demonstrate your final project. If you want to get your final project checked of before finals week, send an email request to set up a demo time.

Final Project Report

> ⚠ These requirements are not finalized yet...

A project report **final-project.md** needs to be submitted to the *final-project* repository in the \docs folder. Your final project report needs to have the following sections:

- **Introduction** Give an overview of your system.

- **Body**

  1. Describe the hardware and software components that you used.
  2. Create a section and write a description of what the hardware and software components do. Give a description of each VHDL process in your hardware components. Give a description of each function in your software.
  3. Create a section on your custom hardware/software that you proposed.

- **Conclusion** Make a conclusion and relate your experience regarding the creation and execution of your project. What would you improve regarding the course?

Project Files and Project Submission

Your *final-project* repository will serve as the location for your final project files submission. *To submit your final project*, create a **tag** that says *final-project-submission*. Your source code should be your own work and should be well commented.